

Crime Rate in India and How can it affect tourism & Public Safety?

Anandkarthick

Abstract

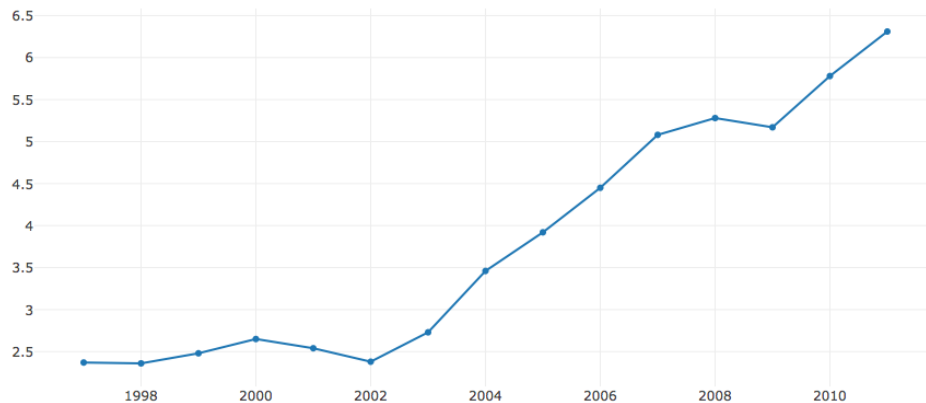
As part of the project, I used "Crime in India" dataset from Kaggle. Research questions were to Identify prominent tourist spots in India and to understand if 'Murder' & 'Rape' occurs in that state along with studying police force available for support.

I used Pandas Numpy methods along with Plotly methods to analyze and represent the findings visually.

Crime rate is high in many states and there is not enough police force to ensure public safety. Most of the states have prominent tourist spots, which could create a challenge to tourism if not noticed.

Motivation

Average Number of Visitors to India - 1997 to 2011



Tourism in India is economically important and is growing rapidly. The World Travel & Tourism Council calculated that tourism generated ₹14.02 lakh crore (USD 220 billion) or 9.6% of the nation's GDP in 2016 and supported 40.343 million jobs, 9.3% of its total employment. The sector is predicted to grow at an annual rate of 6.8% to ₹28.49 lakh crore (USD 440 billion) by 2027 (10% of GDP).

In October 2015, India's medical tourism sector was estimated to be worth USD 3 billion. It is projected to grow to USD 7–8 billion by 2020. In 2014, 184,298 foreign patients traveled to India to seek medical treatment.

Recently, India is named in the list of "unsafe" countries for Women travelers and also generally for tourists. Here is the [link](#) to the article

Note - this is a broader question. But the motivation here is to try and understand as much as we can that will help me and others to continue to work on this dataset. As a country, India has a huge potential for tourists and travelers. from around the world.

Dataset(s)

Dataset Credits

- Main Dataset - <https://www.kaggle.com/rajanand/crime-in-india>
 - This is a huge data set. Consists of different crimes (district level), fraud, kidnapping, Juvenile crimes, arrests, police strength etc.
- Supporting datasets (trends, pie charts) - https://en.wikipedia.org/wiki/Tourism_in_India
 - These are supporting data sets created for this project.
 - Latitude and Longitude values are generated via Google API (included in the notebook)

Data Preparation and Cleaning

- Used Tabula to create data file out of PDFs.
- Used Pandas Numpy libraries extensively for aggregation, grouping of the data.
- Used libraries like requests and os to create reusable functions to create location dataset.

Research Question(s)

- 1) What are the prominent tourist spots that attracts visitors from foreign countries?
- 2) How many crimes (Rape or murder) happened in last decade in Indian States? How close are they to the tourist spot?
(visually represented)
- 3) do they (Indian States) have adequate police force to handle and provide support and ensure public safety?

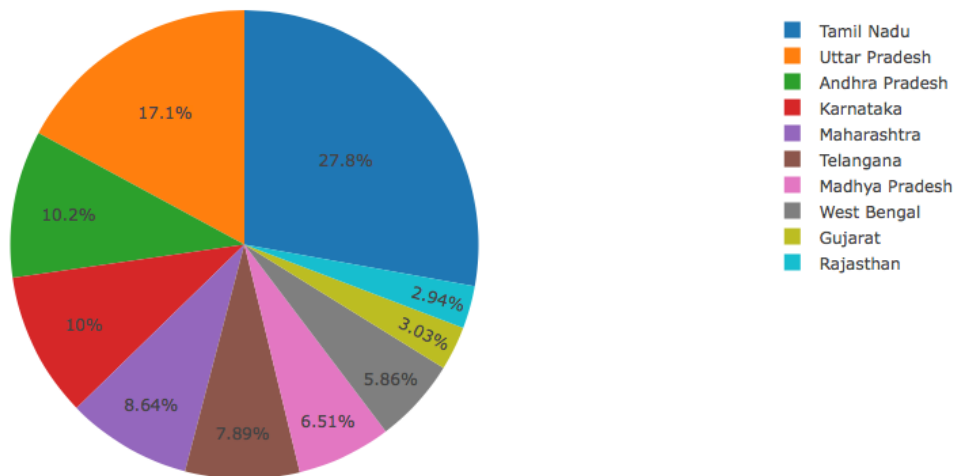
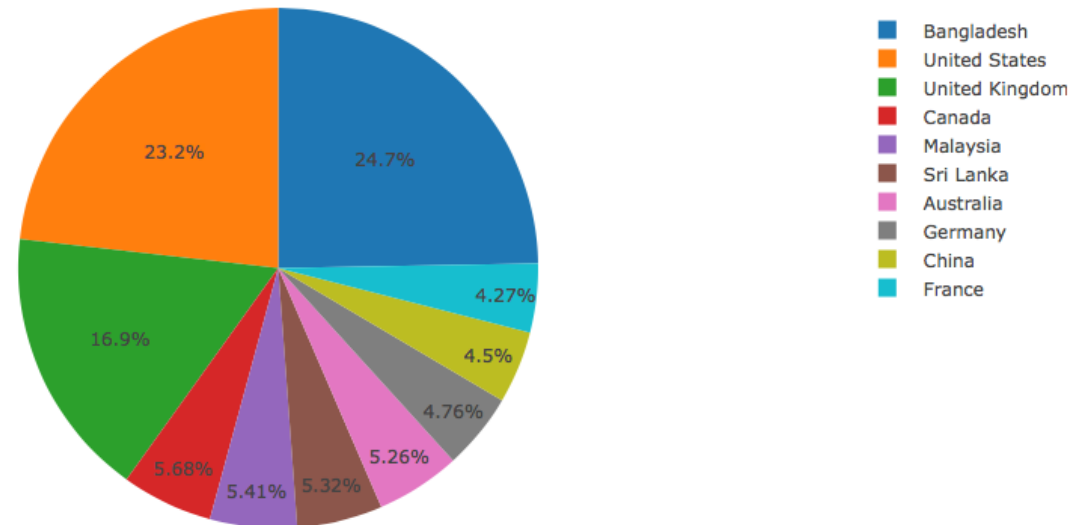
Methods

- Linux commands – ls, cat used to view and locate the source files
- Pandas Numpy libraries along with shape and head methods are used to understand the structure of the data and sample.
- Info is used to understand Index for the data frame.
- Concat method is used to join two data frames before creating visualization.
- Bar, Pie and other methods are used with Plotly to create visuals.

Findings

What's this? - Source countries for foreign tourist arrivals in India

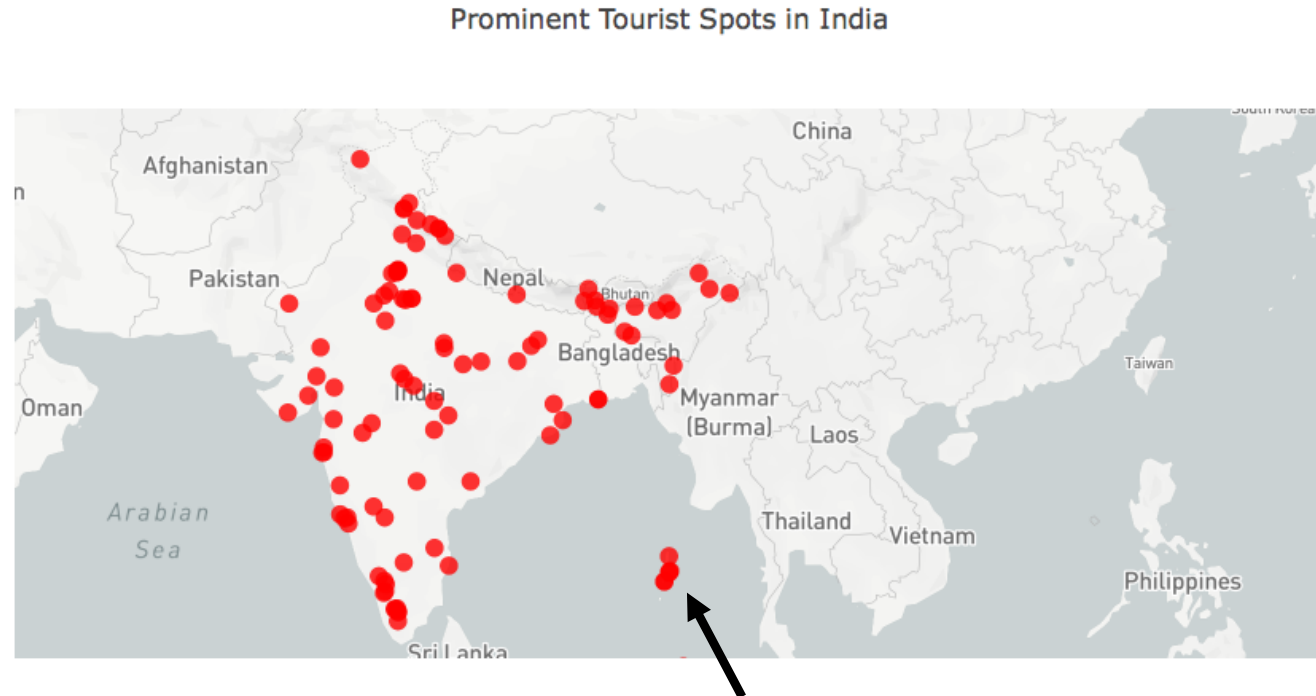
Source – Wikipedia



What's this? - Top 10 states that attracted tourists in 2015.

Source – Wikipedia

What is this? – Prominent Tourist spots plotted with Plotly.

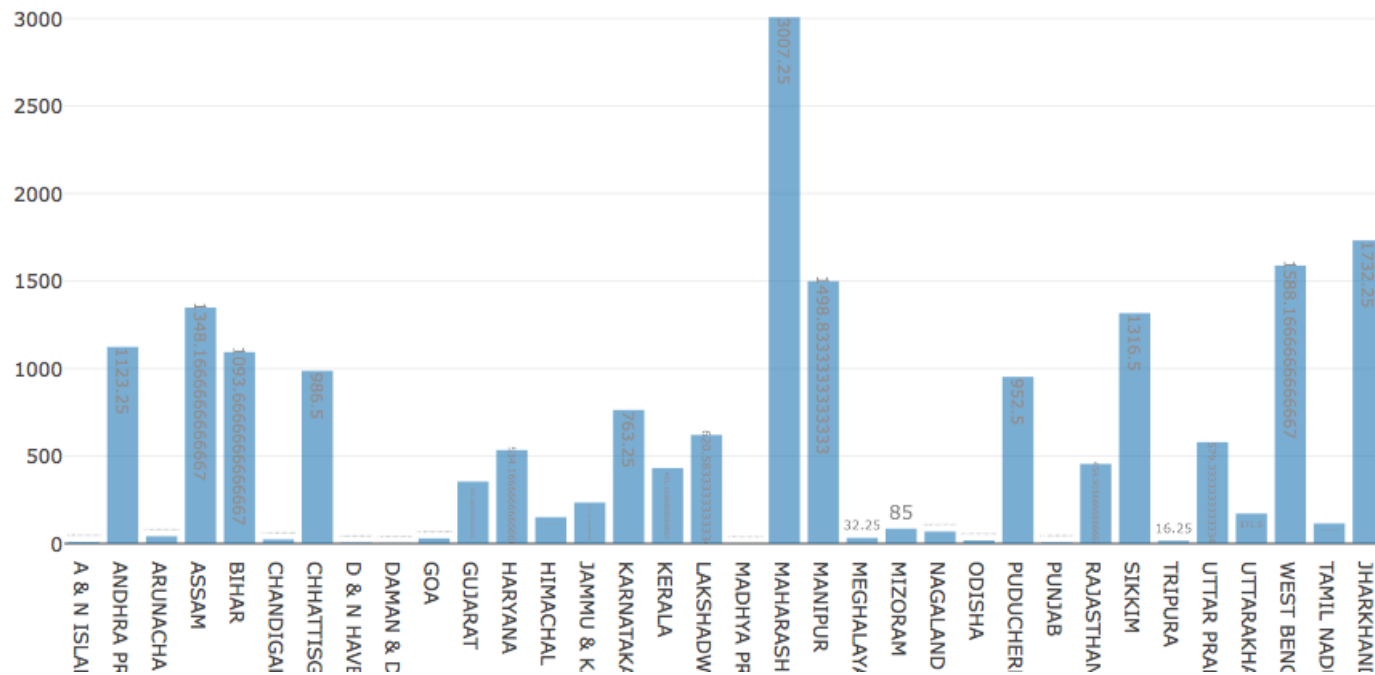


Prominent Tourist Spots in India

- UNESCO world heritage sites.
- Popular sites in each state (Source – Wikipedia).

Observations

- Spots are across different state and vastly distributed.
- Andaman & Nicobar – is a very popular destination for a small place (area).



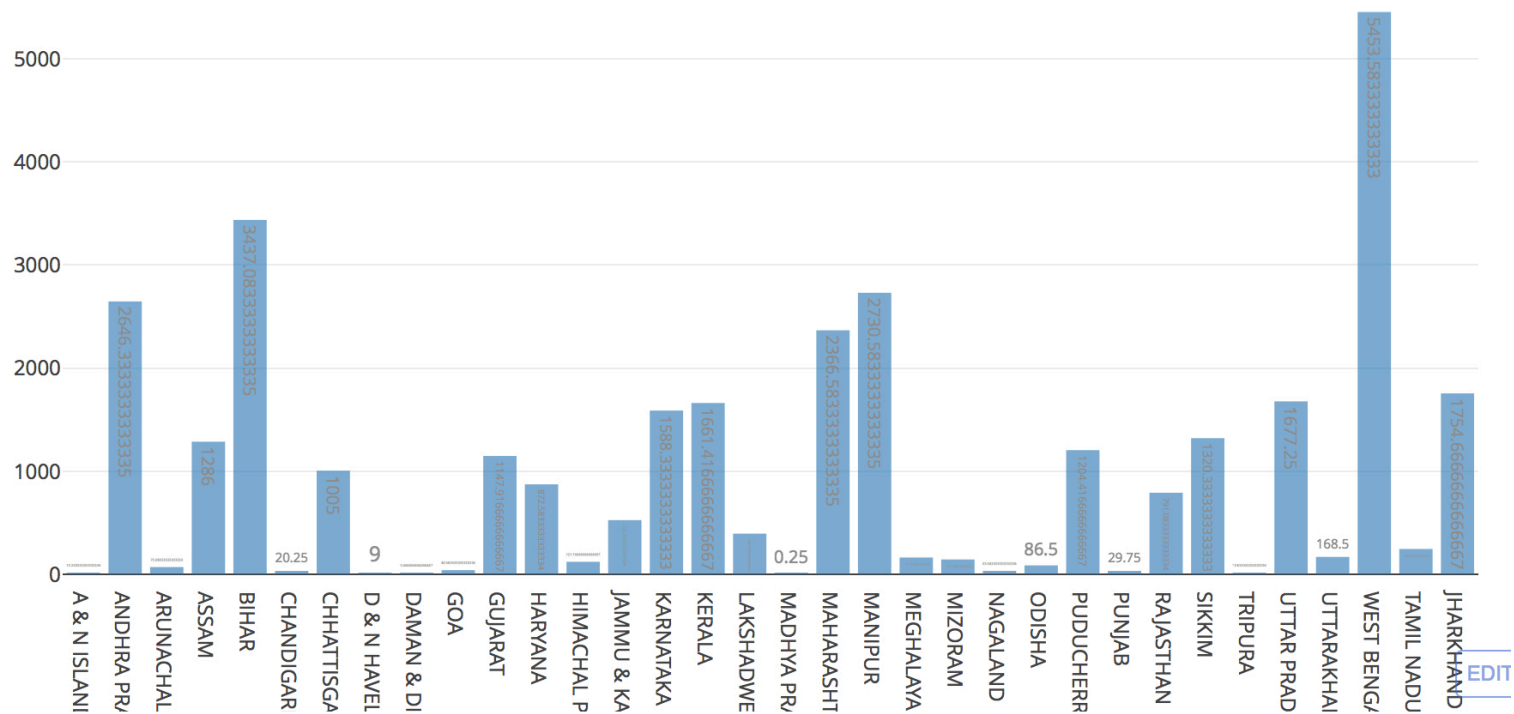
What is this? – Bar representation of one of the indicators ('RAPE').

X axis – States

Y axis – Average Value of the indicator (RAPE)

Observations

- High Spikes (top in the tourist visitor list – Slide 8) for the states **West Bengal**, **Maharashtra**,
- High Spikes for the states **Jharkhand**, **Manipur**. They are not in top 10 (slide 8)
- **Tamil Nadu** (top in the tourist visitor list – Slide 8) has a relatively smaller spike.



What is this? – Bar representation of one of the indicators ('MURDER').

X axis – States

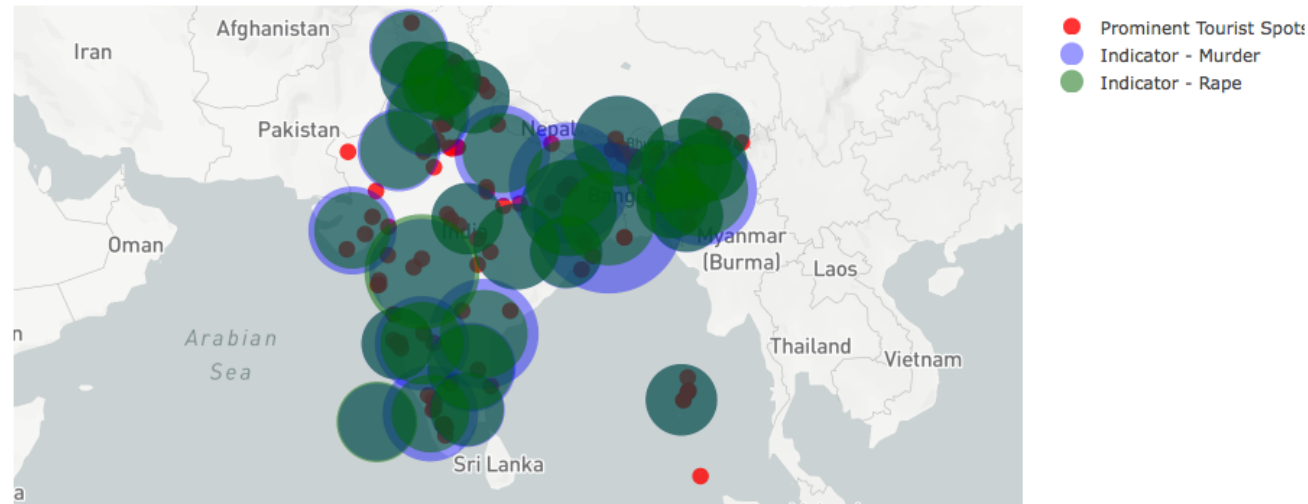
Y axis – Average Value of the indicator (MURDER)

Observations

- High Spikes (top in the tourist visitor list – Slide 8) for the states **West Bengal**, **ANDRA PRADESH**,
- High Spikes for the states **Jharkhand**, **Manipur**. They are not in top 10 (slide 8)
- **Tamil Nadu** (top in the tourist visitor list – Slide 8) has a relatively smaller spike.



Prominent Tourist Spots in India - Along with Indicators

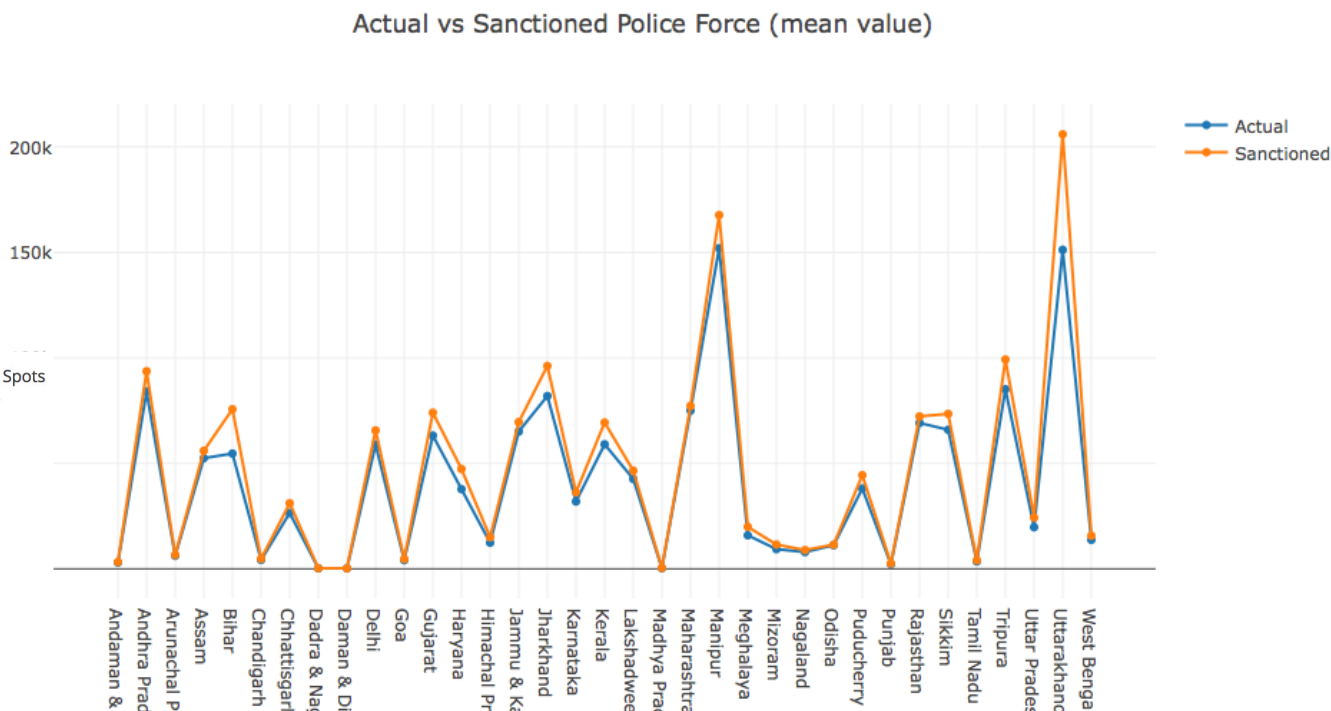
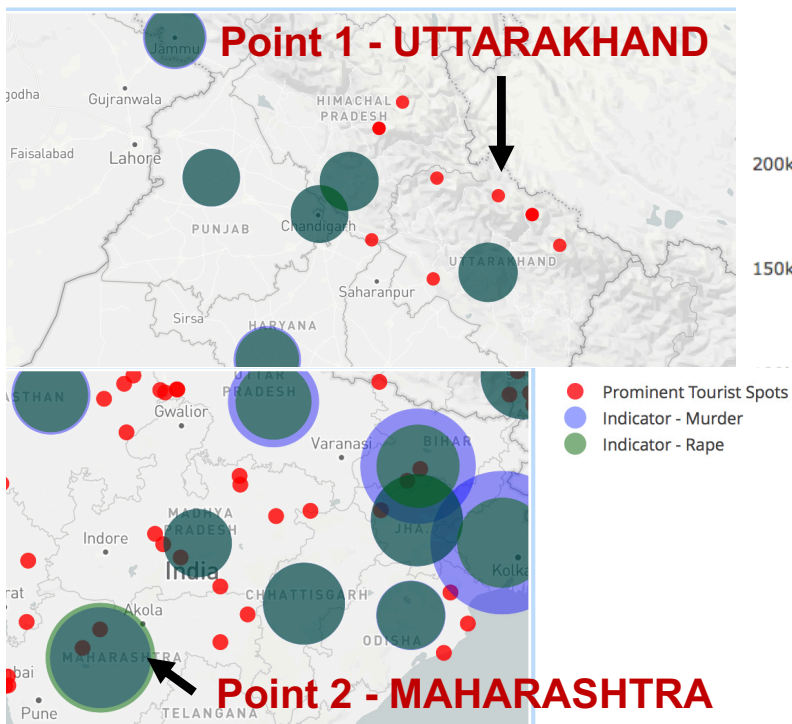


What is this? – Prominent Tourist spots plotted along with Crime Indicators with Plotly.

Indicators – Rape, Murder

Observations

- Very dense circles for the indicator – Rape (green) overall.
- Eastern States have bigger blue circles (Indicator – Murder)
- Northern & Southern states have denser circles for the indicator – Rape.



What is this? – Line plot for Actual vs Sanctioned values of police force for Indian States (mean value).

X axis – States

Y axis – Average Value of the Actual & Sanctioned Police Force.

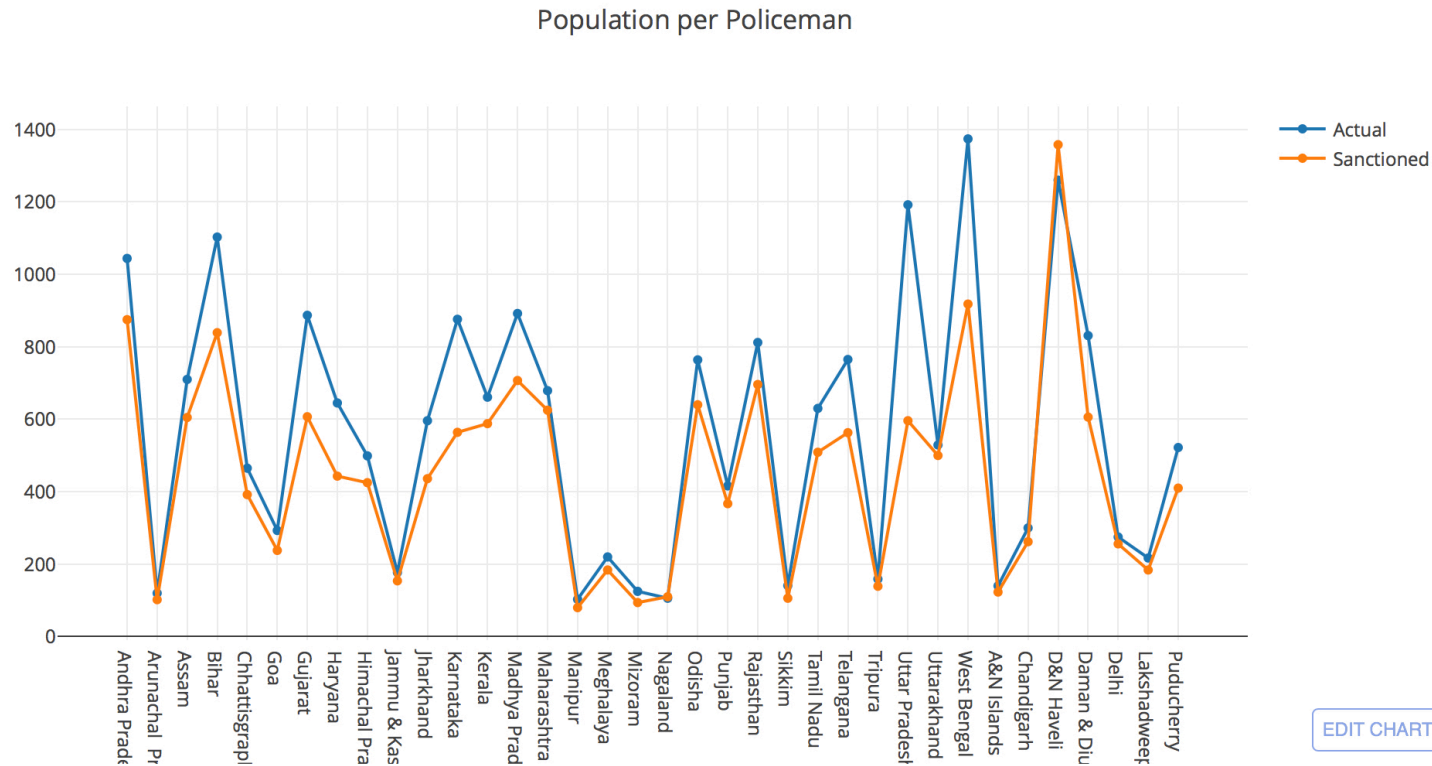
Observations

- Actual vs Sanctioned differences are high for states – **UTTARAKHAND**. This state has more tourist spots. But, looking at the map to the left, this area is also hilly, which could be a contributing factor for the difference in Actual vs Sanctioned.
- MAHARASHTRA** – State has almost no difference but the crime rate is very high

What is this? – Line plot for Actual vs Sanctioned values of population per policeman (mean value).

X axis – States

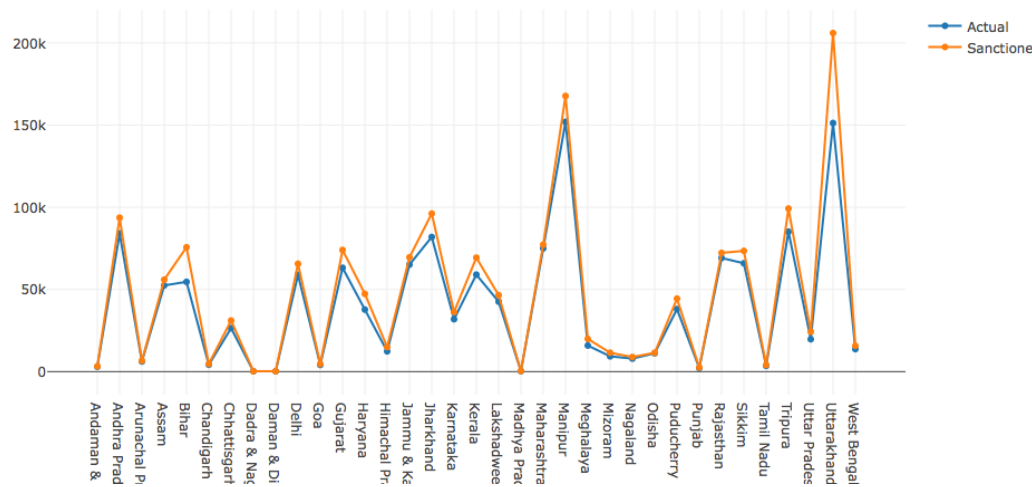
Y axis – Average Value of the Actual & Sanctioned population per policeman.



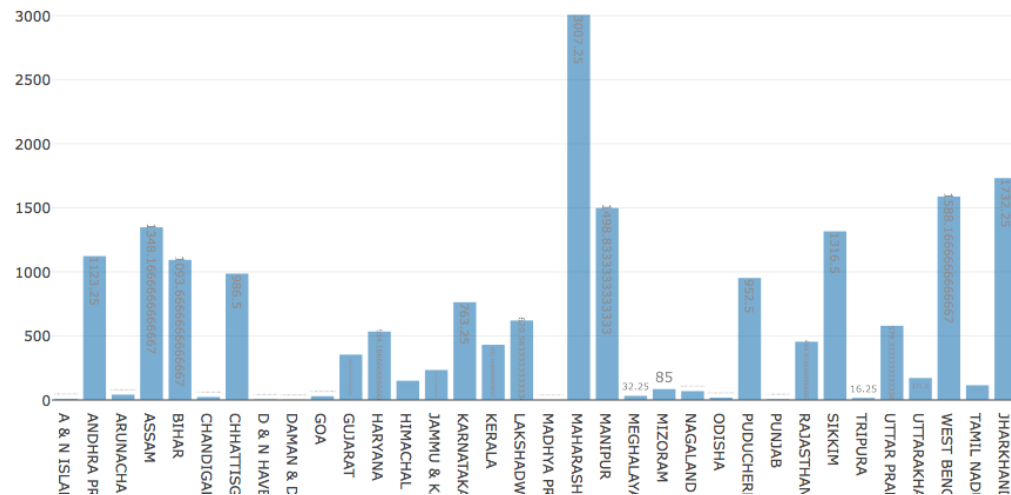
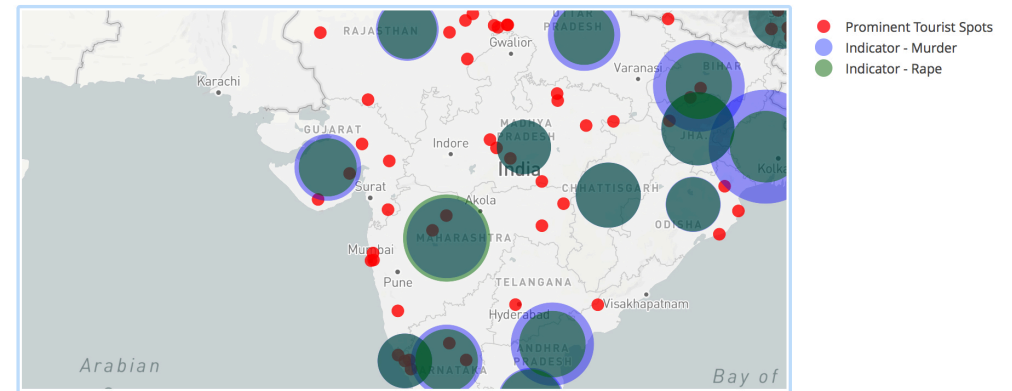
Observations

- Overall, almost states have difference between Actual and Sanctioned. This definitely indicates the need to reevaluate how the police force is distributed for public safety.
- Actual vs Sanctioned differences are high for states – **WEST BENGAL, UTTAR PRADESH**
- For the place **D&N Haveli**, the sanctioned is high. This place has very minimal number in crime (slide 10)

Actual vs Sanctioned Police Force (mean value)



Prominent Tourist Spots in India - Along with Indicators



Observations

- Police force is not distributed properly to ensure public safety.
- Crime rates are high in many states that have prominent tourist spots.
- We have seen a variety of combinations. Certain states with high police force have higher crime rates and vice versa.
- There is a need for government to understand the gravity of the issue.

Limitations

Though these observations tell us about the threat to Indian tourism industry, it doesn't have all other factors and data to give a holistic picture. For example, we can correlate this data with education to understand how it impacts on crime.

Urban vs Rural indicators will help us understand and separate the exact regions of the crime. We don't have enough data to relate and conclude.

Conclusions

- 1) What are the prominent tourist spots that attracts visitors from foreign countries?
 - **Prominent tourist spots are present across different states and districts. India has UNESCO sites and also popular temples, churches. For any tourist who comes to visit the country will have a longer stay to cover multiple places.**

- 2) How many crimes (Rape or murder) happened in last decade in Indian States? How close are they to the tourist spot? (visually represented)
 - **Between 2000 and 2012, we have seen over 1500 RAPES, Murders on average for certain states with tourist spots.**
 - **West Bengal (State) stands at the top for both indicators and it is one of the most visit placed but it doesn't have prominent tourist spot listed.**

- 3) do they (Indian States) have adequate police force to handle and provide support and ensure public safety?
 - **Not really. We have seen People to Policeman trend and it's staggering. Almost all states have difference between actual and sanctioned.**

Acknowledgements

Thanks to

- Kaggle (dataset)
- Wikipedia (reference data, story, motivation)
- Google (location data)
- Stack Overflow (for everything)
- Mapbox, plotly (amazing visualization)
- Edx and UcSanDiegoX (superb notes and reference videos)
- Tabula (for creating datasets from pdfs)
- Wife (for letting me do this course and support)

References

- Wikipedia (reference data, story, motivation)
- Google (location data)
- Stack Overflow (for everything)
- Mapbox, plotly (amazing visualization and documents)
- Edx and UcSanDiegoX (superb notes and reference videos)

Crime Rate in India and How can it affect tourism & Public Safety?

Tourism in India is economically important and is growing rapidly. The World Travel & Tourism Council calculated that tourism generated ₹14.02 lakh crore (USD 220 billion) or 9.6\% of the nation's GDP in 2016 and supported 40.343 million jobs, 9.3\% of its total employment.

The sector is predicted to grow at an annual rate of 6.8% to ₹28.49 lakh crore (USD 440 billion) by 2027 (10% of GDP). In October 2015, India's medical tourism sector was estimated to be worth USD 3 billion. It is projected to grow to USD 7–8 billion by 2020. In 2014, 184,298 foreign patients traveled to India to seek medical treatment.

Recently, India is named in the list of "unsafe" countries for Women travelers and also generally for tourists. Here is the link to the article

[Follow the link here \(https://www.forbes.com/sites/laurabegleybloom/2017/07/28/10-most-dangerous-places-for-women-travelers-and-how-to-stay-safe/#2b4872eb2448\)](https://www.forbes.com/sites/laurabegleybloom/2017/07/28/10-most-dangerous-places-for-women-travelers-and-how-to-stay-safe/#2b4872eb2448)

Dataset Credits

- Main Dataset - <https://www.kaggle.com/rajanand/crime-in-india> (<https://www.kaggle.com/rajanand/crime-in-india>)
- Supporting datasets (trends, pie charts) - https://en.wikipedia.org/wiki/Tourism_in_India (https://en.wikipedia.org/wiki/Tourism_in_India)
- Introduction Source - https://en.wikipedia.org/wiki/Tourism_in_India (https://en.wikipedia.org/wiki/Tourism_in_India)

Visualization Credits

- Plotly - <https://plot.ly> (<https://plot.ly>)
 - Mapbox - <https://www.mapbox.com/> (<https://www.mapbox.com/>)
- </html>

```
In [815]: import pandas as pd
import requests
import os
import matplotlib.pyplot as plt
import plotly
plotly.tools.set_credentials_file(username=' ', api_key=' ')
import plotly.plotly as py
from plotly.graph_objs import *
```

```
In [621]: # Let's see some trends and numbers to understand the motivation - Source (Wikipedia)

tour_trend = pd.read_csv('./crime-in-india/external/India_tourism_trend.csv', encoding='latin-1')
tour_trend.head()
```

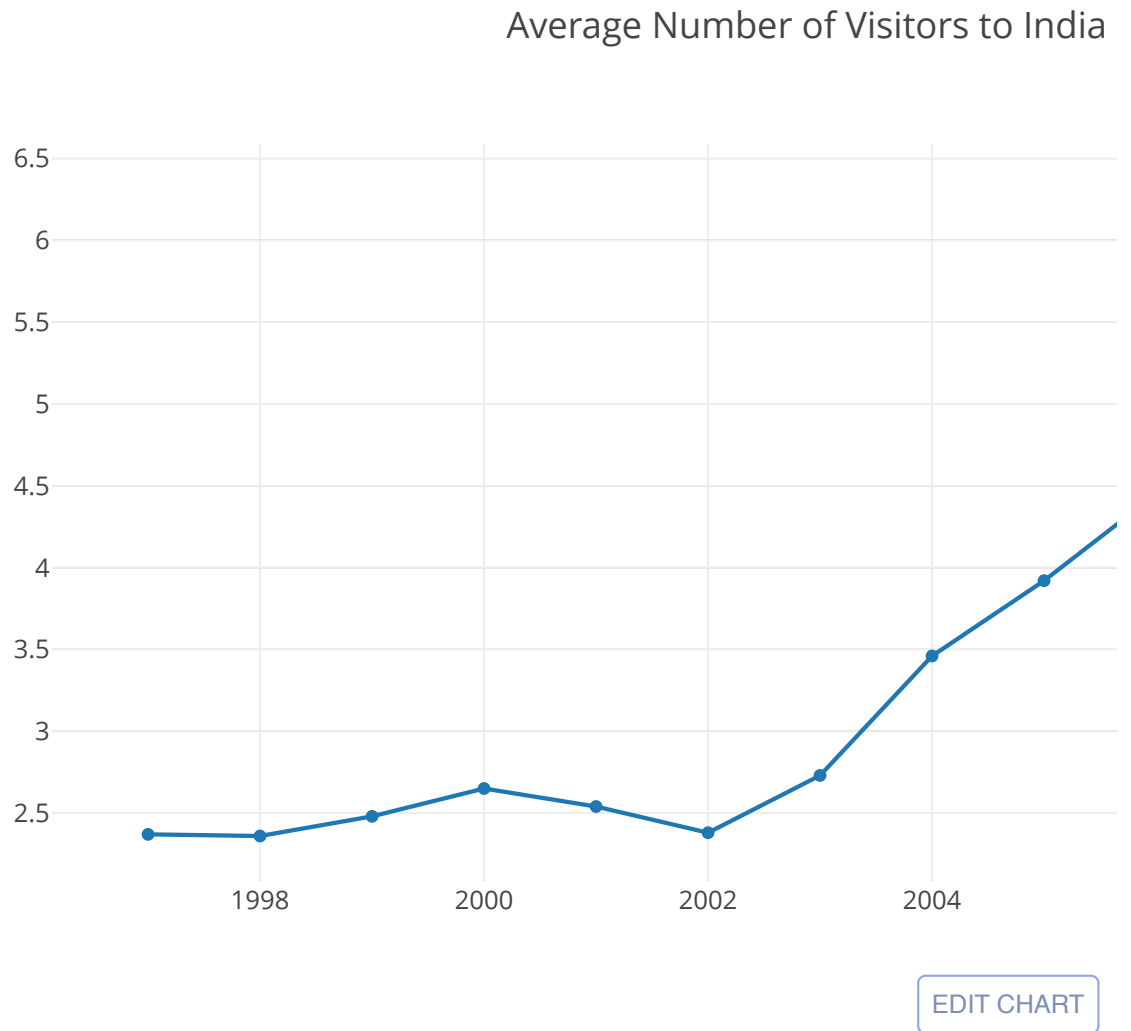
Out[621]:

	Year	Number (millions)	æ% change
0	1997	2.37	3.8
1	1998	2.36	-0.7
2	1999	2.48	5.2
3	2000	2.65	6.7
4	2001	2.54	-4.2

```
In [622]: # Plotting the trend using plotly

val_x = tour_trend.Year
val_y = tour_trend['Number (millions)']
trace = Scatter(
    x = val_x,
    y = val_y,
    mode = 'lines+markers',
    name = 'lines+markers')
data = [trace]
layout = dict(title='Average Number of Visitors to India - 1997 to 2011')
fig = dict(data=data, layout=layout)
#trace = Scatter(x,y)
py.iplot(fig, file_name='line')
```

Out[622]:



Average visitors is growing despite few dips in between. India has a culturally rich places, forts, temples, churches to visit. Unesco has almost 36 heritage sites across the country with 25 more pending to be approved. We will see some prominent tourist spots as we progress.

```
In [623]: # Source countries for foreign tourist arrivals in India : top 10 countries. Source (Wikipedia).

tour_trend = pd.read_csv('./crime-in-india/external/Foreign_Visitors.csv',encoding='latin-1')
tour_trend.head()
```

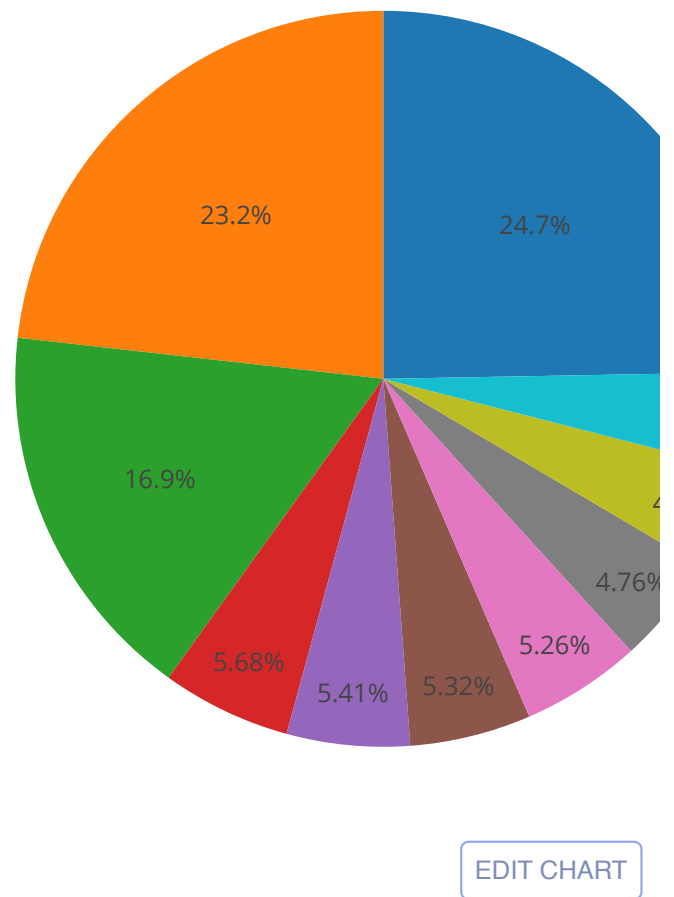
Out[623]:

	Country	Visitors
0	Bangladesh	1.380409
1	United States	1.296939
2	United Kingdom	0.941883
3	Canada	0.317239
4	Malaysia	0.301961

```
In [626]: # Plotting the pie to visually see the number

labels = tour_trend.Country
values = tour_trend.Visitors
trace = Pie(labels=labels, values=values)
py.iplot([trace], title='basic_pie_chart')
```


Out[626]:



Bangladesh is very close to India and apparently is at top of the list. Chart is diverse and has countries from several continents

```
In [628]: # Share of top 10 states/UTs of India in number of domestic tourist vi
sits. Source (Wikipedia)

tour_rev = pd.read_csv('./crime-in-india/external/top_tourist_visiting
_places.csv')
tour_rev.head()
```

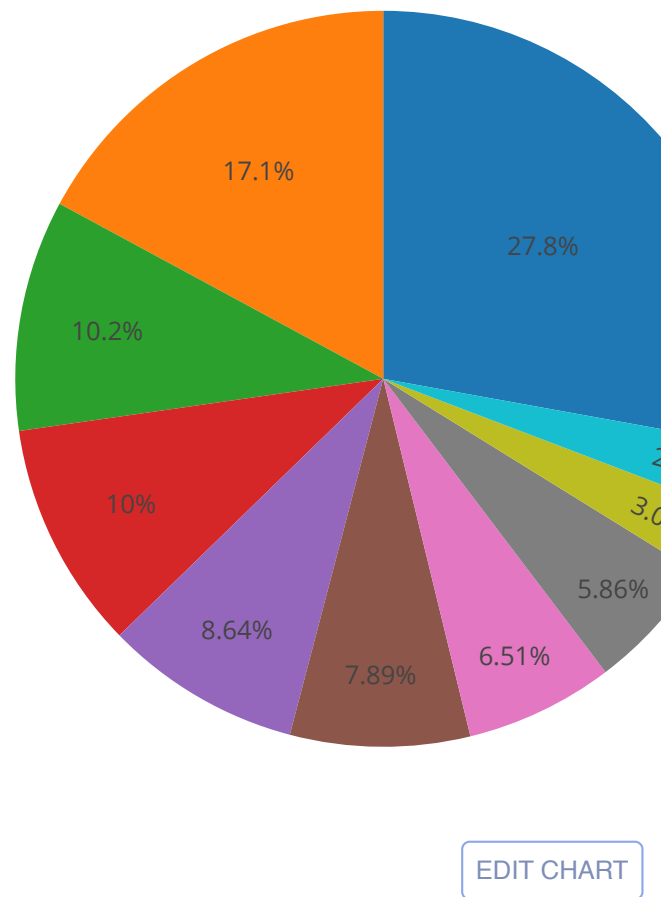
Out[628]:

	State	Tourists_in_Millions	Tourists
0	Tamil Nadu	333.459047	333,459,047
1	Uttar Pradesh	204.888457	204,888,457
2	Andhra Pradesh	121.591054	121,591,054
3	Karnataka	119.863942	119,863,942
4	Maharashtra	103.403934	103,403,934

```
In [630]: # Plotting pie chart

labels = tour_rev.State
values = tour_rev.Tourists_in_Millions
trace = Pie(labels=labels, values=values)
py.iplot([trace], title='basic_pie_chart')
```

Out[630]:



Most of these places are covered in the bottom section, where we discuss about crime, tourists spots.

```
In [1]: ! ls ../crime-in-india/external
```

```
API_EG.ELC.ACCS.ZS_DS2_en_csv_v2.zip
Chapter_26_Tourism-Table_26.2-Rankwise_Calculation.csv
NO_OF_APPROVED_ACCOMODATION_AND_ROOMS_ALONG_WITH_THOSE_PENDING_FOR_C
LASSIFICATION.xls
National_Parks_India.csv
National_Parks_India.xlsx
Tourism_Statistics_Since_1991.xls
rs_session-239_AU693_1.1.csv
whc-sites-2017.xls
```

```
In [631]: ### Let's start by identifying prominent tourist spots in India.  
# This list has UNESCO sites and other popular places from Wikipedia  
  
National_Sites = pd.read_csv('./crime-in-india/external/np_india.csv',  
encoding='latin-1')  
National_Sites.head()  
National_Sites.shape
```

Out[631]: (111, 4)

```
In [486]: # Function to get Latitude and Longitude locations for any given address or name.  
# this is reusable and we use it more than once as we progress.  
  
def get_lat_long(xname, yname):  
    temp_list = []  
    response = requests.get('https://maps.googleapis.com/maps/api/geocode/json?address=' + xname.lower() + '')  
    resp_json_payload = response.json()  
    if ((resp_json_payload['status']) == 'OK'):  
        temp_list = list(dict.values(resp_json_payload['results'][0]['geometry']['location']))  
        temp_list.append(yname)  
        return temp_list  
    else:  
        temp_list = []  
        return temp_list
```

```
In [157]: # So, we don't have latitude and longitude information for the tourist spots.  
# Let's create a document with spots and lat/long information.  
  
if os.path.exists('./crime-in-india/external/np_lat_long.csv'):  
    os.remove('./crime-in-india/external/np_lat_long.csv')  
  
temp_list = list(National_Sites['Name'])  
with open('./crime-in-india/external/np_lat_long.csv', 'w') as f:  
    f.write("lat,lon,place")  
    f.write('\n')  
    for i in range(len(temp_list)):  
        temp_str = temp_list[i].replace(" ", "+")  
        line = str(get_lat_long(temp_str, temp_list[i]))  
        jk_line1 = line.replace("[", "")  
        jk_line2 = jk_line1.replace("]", "")  
        f.write(jk_line2)  
        f.write('\n')
```

```
In [235]: # Exploring the output document from above step. Now, we have lat/long  
information to plot.  
  
tframe = pd.read_csv('./crime-in-india/external/np_lat_long.csv')  
tframe.shape
```

```
Out[235]: (104, 3)
```

```
In [963]: # Using Plotly, let's plot the tourist spots.
# We can zoom in and out to interact with the visual.

mapbox_access_token = ' '

df = pd.read_csv('./crime-in-india/external/np_lat_long.csv')
site_lat = df.lat
site_lon = df.lon
locations_name = df.place

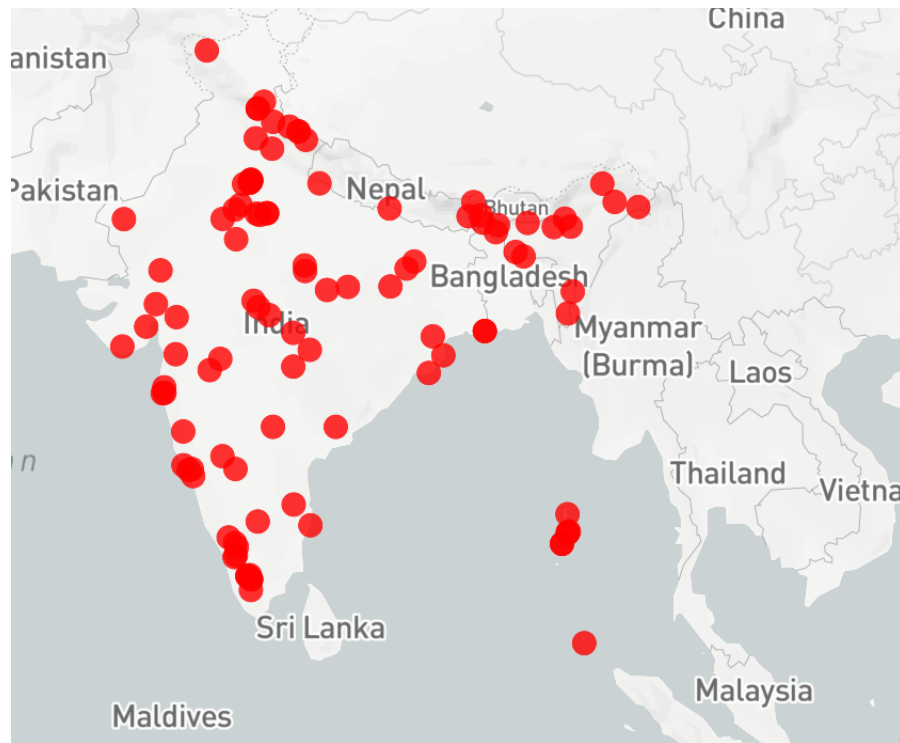
data = Data([
    Scattermapbox(
        lat=site_lat,
        lon=site_lon,
        mode='markers',
        marker=Marker(
            size=12,
            color='rgb(255, 0, 0)',
            opacity=0.8
        ),
        text=locations_name,
        hoverinfo='text'
    )
])

layout = Layout(
    title='Prominent Tourist Spots in India',
    autosize=True,
    hovermode='closest',
    showlegend=False,
    mapbox=dict(
        accesstoken=mapbox_access_token,
        bearing=0,
        center=dict(
            lat=22.6,
            lon=93.4
        ),
        pitch=0,
        zoom=3,
        style='light'
    ),
)

fig = dict(data=data, layout=layout)
py.iplot(fig, filename='India Mapbox')
```

Out[963]:

Prominent Tourist Spots in India

[EDIT CHART](#)

So, this is vast and all over. The important thing is, these are not the only spots. Infact, we didn't include temples and churches in list Individually. But, those spots are covered in the data.

Identifying & Plotting Crime Indicators

```
In [949]: crime_data = pd.read_csv('./crime-in-india/Crime_data/01_District_wise
_crimes_committed_IPC_2001_2012.csv')
crime_data.head()
```

Out[949]:

	STATE/UT	DISTRICT	YEAR	MURDER	ATTEMPT TO MURDER	CULPABLE HOMICIDE NOT AMOUNTING TO MURDER	RAPE	CUSTOD R/
0	ANDHRA PRADESH	ADILABAD	2001	101	60	17	50	0
1	ANDHRA PRADESH	ANANTAPUR	2001	151	125	1	23	0
2	ANDHRA PRADESH	CHITTOOR	2001	101	57	2	27	0
3	ANDHRA PRADESH	CUDDAPAH	2001	80	53	1	20	0
4	ANDHRA PRADESH	EAST GODAVARI	2001	82	67	1	23	0

5 rows × 33 columns

```
In [633]: # Let's understand the data range
```

```
crime_data['YEAR'].unique().tolist()
```

```
Out[633]: [2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012]
```

```
In [635]: # Understanding the number of states.
```

```
sample_state = crime_data['STATE/UT'].unique().tolist()
len(sample_state)
```

```
Out[635]: 35
```



```
In [639]: # Data is very granular and present at district level.  
# Grouping the data on State/UT level to understand the size of the data.  
  
crime_data[['STATE/UT', 'YEAR']].groupby('STATE/UT').count().sort_values('YEAR', ascending=False)
```

```
In [224]: # Every state has a district "Total" which actually indicates sum of values.  
# Here, we play around few filters to get a feel for the data we are dealing with.  
  
#filter = crime_data['STATE/UT'] == 'LAKSHADWEEP'  
filter = (crime_data['STATE/UT'] == 'LAKSHADWEEP') & (crime_data['DISTRICT'] == 'TOTAL')  
crime_data[filter]
```

Out[224]:

	STATE/UT	DISTRICT	YEAR	MURDER	ATTEMPT TO MURDER	CULPABLE HOMICIDE NOT AMOUNTING TO MURDER	RAPE	CL
713	LAKSHADWEEP	TOTAL	2001	1	0	0	0	0
1432	LAKSHADWEEP	TOTAL	2002	0	0	0	1	0
2160	LAKSHADWEEP	TOTAL	2003	0	0	0	2	0
2889	LAKSHADWEEP	TOTAL	2004	0	0	0	0	0
3622	LAKSHADWEEP	TOTAL	2005	1	0	0	0	0
4361	LAKSHADWEEP	TOTAL	2006	0	2	0	0	0
5104	LAKSHADWEEP	TOTAL	2007	0	2	0	1	0
5865	LAKSHADWEEP	TOTAL	2008	1	0	0	2	0
6632	LAKSHADWEEP	TOTAL	2009	0	1	0	1	0
7411	LAKSHADWEEP	TOTAL	2010	0	0	0	0	0
8202	LAKSHADWEEP	TOTAL	2011	0	1	0	0	0
9013	LAKSHADWEEP	TOTAL	2012	0	0	0	0	0

12 rows x 33 columns

In [646]: *# Filtering out the district "Total" from the dataframe.*

```
t_filter = crime_data['DISTRICT'] == 'TOTAL'
new_sample = crime_data[t_filter]
```

In [642]: *# Grouping should give exactly 12 counts for each state. Depending on that, we will do more clean up for the frame.*

```
new_sample[['STATE/UT', 'YEAR']].groupby('STATE/UT').count().sort_values('YEAR', ascending=False)
```

Out[642]:

--	--

	YEAR
STATE/UT	
A & N ISLANDS	12
PUDUCHERRY	12
MAHARASHTRA	12
MANIPUR	12
MEGHALAYA	12
MIZORAM	12
NAGALAND	12
ODISHA	12
PUNJAB	12
ANDHRA PRADESH	12
RAJASTHAN	12
SIKKIM	12
TAMIL NADU	12
TRIPURA	12
UTTAR PRADESH	12
UTTARAKHAND	12
MADHYA PRADESH	12
LAKSHADWEEP	12
KERALA	12
KARNATAKA	12
JHARKHAND	12
JAMMU & KASHMIR	12
HIMACHAL PRADESH	12
HARYANA	12
GUJARAT	12
GOA	12
DAMAN & DIU	12

D & N HAVELI	12
CHHATTISGARH	12
CHANDIGARH	12
BIHAR	12
ASSAM	12
ARUNACHAL PRADESH	12
WEST BENGAL	12

```
In [648]: # Looking at indicators in the dataset.
```

```
list(crime_data)
```

```
Out[648]: ['STATE/UT',  
          'DISTRICT',  
          'YEAR',  
          'MURDER',  
          'ATTEMPT TO MURDER',  
          'CULPABLE HOMICIDE NOT AMOUNTING TO MURDER',  
          'RAPE',  
          'CUSTODIAL RAPE',  
          'OTHER RAPE',  
          'KIDNAPPING & ABDUCTION',  
          'KIDNAPPING AND ABDUCTION OF WOMEN AND GIRLS',  
          'KIDNAPPING AND ABDUCTION OF OTHERS',  
          'DACOITY',  
          'PREPARATION AND ASSEMBLY FOR DACOITY',  
          'ROBBERY',  
          'BURGLARY',  
          'THEFT',  
          'AUTO THEFT',  
          'OTHER THEFT',  
          'RIOTS',  
          'CRIMINAL BREACH OF TRUST',  
          'CHEATING',  
          'COUNTERFIETING',  
          'ARSON',  
          'HURT/GREVIOUS HURT',  
          'DOWRY DEATHS',  
          'ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY',  
          'INSULT TO MODESTY OF WOMEN',  
          'CRUELTY BY HUSBAND OR HIS RELATIVES',  
          'IMPORTATION OF GIRLS FROM FOREIGN COUNTRIES',  
          'CAUSING DEATH BY NEGLIGENCE',  
          'OTHER IPC CRIMES',  
          'TOTAL IPC CRIMES']
```

```
In [644]: # for this project, let's start with "Murder" & "Rape" indicators.  
  
temp_crime_data = crime_data[t_filter]  
  
t_crime_data = temp_crime_data[['STATE/UT', 'YEAR', 'MURDER', 'RAPE']]  
t_crime_data.head()
```

Out[644]:

	STATE/UT	YEAR	MURDER	RAPE
28	ANDHRA PRADESH	2001	2602	871
42	ARUNACHAL PRADESH	2001	83	33
70	ASSAM	2001	1356	817
115	BIHAR	2001	3643	888
135	CHHATTISGARH	2001	880	959

```
In [652]: # exploring the data frame further to understand the value distributio
n, correlation.

print("Exploring values for the indicator - RAPE")
list1 = t_crime_data['RAPE'].unique().tolist()
print("Range --> Min Value : " + str(min(list1)) + " to " + str(max(list1)) )
print("Mean Value - " + str(t_crime_data['RAPE'].mean()))
print("Median Value - " + str(t_crime_data['RAPE'].median()))
print("Mode Value - " + str(t_crime_data['RAPE'].mode().values))
print("\n")
print("Exploring values for the indicator - MURDER")
list2 = t_crime_data['MURDER'].unique().tolist()
print("Range --> Min Value : " + str(min(list2)) + " to " + str(max(list2)) )
print("Mean Value - " + str(t_crime_data['MURDER'].mean()))
print("Median Value - " + str(t_crime_data['MURDER'].median()))
print("Mode Value - " + str(t_crime_data['MURDER'].mode().values))

Corr1 = t_crime_data['RAPE'].corr(temp_crime_data['ATTEMPT TO MURDER'] )
print(Corr1)
Corr2 = t_crime_data['RAPE'].corr(temp_crime_data['ASSAULT ON WOMEN WITH INTENT TO OUTRAGE HER MODESTY'])
print(Corr2)
```

```
Exploring values for the indicator - RAPE
Range --> Min Value : 0 to 3425
Mean Value - 570.375
Median Value - 281.5
Mode Value - [0 1]
```

```
Exploring values for the indicator - MURDER
Range --> Min Value : 0 to 7601
Mean Value - 971.4166666666666
Median Value - 381.0
Mode Value - [0]
0.68290373222
0.856239698998
```

```
In [405]: t_crime_data.sort_values('STATE/UT')
```

```
Out[405]:
```

	STATE/UT	YEAR	MURDER	RAPE
7384	A & N ISLANDS	2010	9	24
8986	A & N ISLANDS	2012	15	12

1409	A & N ISLANDS	2002	17	2
691	A & N ISLANDS	2001	13	3
5838	A & N ISLANDS	2008	8	12
5081	A & N ISLANDS	2007	15	3
2867	A & N ISLANDS	2004	15	10
8175	A & N ISLANDS	2011	14	13
3600	A & N ISLANDS	2005	14	4
4339	A & N ISLANDS	2006	4	6
6605	A & N ISLANDS	2009	15	18
2138	A & N ISLANDS	2003	21	2
1464	ANDHRA PRADESH	2003	2667	946
3654	ANDHRA PRADESH	2006	2766	1049
7448	ANDHRA PRADESH	2011	2808	1442
8239	ANDHRA PRADESH	2012	2717	1341
4394	ANDHRA PRADESH	2007	2665	1070
2921	ANDHRA PRADESH	2005	2750	935
28	ANDHRA PRADESH	2001	2602	871
5898	ANDHRA PRADESH	2009	2449	1188
5137	ANDHRA PRADESH	2008	2690	1257
6669	ANDHRA PRADESH	2010	2538	1362
744	ANDHRA PRADESH	2002	2525	1002
2192	ANDHRA PRADESH	2004	2579	1016
3670	ARUNACHAL PRADESH	2006	60	37
7466	ARUNACHAL PRADESH	2011	65	42
758	ARUNACHAL PRADESH	2002	61	38
2937	ARUNACHAL PRADESH	2005	81	35
1480	ARUNACHAL PRADESH	2003	77	31
5915	ARUNACHAL PRADESH	2009	59	59
...

6563	UTTAR PRADESH	2009	4534	1759
5040	UTTAR PRADESH	2007	5000	1648
5796	UTTAR PRADESH	2008	4564	1871
1366	UTTAR PRADESH	2002	6825	1415
8938	UTTAR PRADESH	2012	4966	1963
650	UTTAR PRADESH	2001	7601	1958
8952	UTTARAKHAND	2012	217	148
664	UTTARAKHAND	2001	316	74
6577	UTTARAKHAND	2009	195	111
4311	UTTARAKHAND	2006	274	147
2109	UTTARAKHAND	2003	268	107
5810	UTTARAKHAND	2008	223	87
8146	UTTARAKHAND	2011	178	129
2838	UTTARAKHAND	2004	262	115
5054	UTTARAKHAND	2007	268	117
3571	UTTARAKHAND	2005	279	133
1380	UTTARAKHAND	2002	292	89
7356	UTTARAKHAND	2010	176	121
5835	WEST BENGAL	2008	1811	2263
5079	WEST BENGAL	2007	1652	2106
8982	WEST BENGAL	2012	2252	2046
1406	WEST BENGAL	2002	1444	759
4336	WEST BENGAL	2006	1425	1731
3597	WEST BENGAL	2005	1453	1686
6602	WEST BENGAL	2009	2068	2336
8172	WEST BENGAL	2011	2109	2363
688	WEST BENGAL	2001	1594	709
2135	WEST BENGAL	2003	1425	1002
7381	WEST BENGAL	2010	2398	2311

2864	WEST BENGAL	2004	1425	1475
------	-------------	------	------	------

408 rows × 4 columns

```
In [547]: # So, we have data for 12 years and to plot them let's take a mean of
           # the values.
           # We are creating the marker values for the indicators to plot them in
           # the map.

           plot_crime_data = pd.DataFrame(t_crime_data[['STATE/UT', 'MURDER']].groupby('STATE/UT').mean())
           plot_crime_data['Murder_Marker_Value'] = t_crime_data[['STATE/UT', 'MURDER']].groupby('STATE/UT').mean()/100
           plot_crime_data['RAPE'] = pd.DataFrame(t_crime_data[['STATE/UT', 'RAPE']].groupby('STATE/UT').mean())
           plot_crime_data['Rape_Marker_Value'] = t_crime_data[['STATE/UT', 'RAPE']].groupby('STATE/UT').mean()/100
           plot_crime_data['STATE/UT'] = t_crime_data['STATE/UT'].sort_values().unique()
           plot_crime_data.head()
```

Out[547]:

	MURDER	Murder_Marker_Value	RAPE	Rape_Marker_Value	
STATE/UT					
A & N ISLANDS	13.333333	0.133333	9.083333	0.090833	A & N ISLANDS
ANDHRA PRADESH	2646.333333	26.463333	1123.250000	11.232500	ANDHRA PRADESH
ARUNACHAL PRADESH	70.083333	0.700833	41.666667	0.416667	ARUNACHAL PRADESH
ASSAM	1286.000000	12.860000	1348.166667	13.481667	ASSAM
BIHAR	3437.083333	34.370833	1093.666667	10.936667	BIHAR

```
In [428]: # Checking the dataset

t_crime_data.head()
#txt_f['place'] = pd.DataFrame(t_crime_data['STATE/UT'].sort_values().
unique())
#txt_f.head()
```

Out[428]:

	STATE/UT	YEAR	MURDER	RAPE
28	ANDHRA PRADESH	2001	2602	871
42	ARUNACHAL PRADESH	2001	83	33
70	ASSAM	2001	1356	817
115	BIHAR	2001	3643	888
135	CHHATTISGARH	2001	880	959

```
In [499]: # Sorting the values to plot (data prep)

t_state = t_crime_data['STATE/UT'].unique()
t_state.sort()
t_state
state_list = t_state.tolist()
```

```
In [498]: # We are getting latitude and longitude for all states to plot the ind
icators.
```

```
import time
#temp_list = list(National_Sites['Name'])
with open('./crime-in-india/external/state_ut_lat_lon.csv', 'a') as f:
    f.write("lat,lon,place")
    f.write('\n')
    for i in range(len(state_list)):
        if (temp_str == 'A & N ISLANDS'):
            temp_str = 'Andaman Nicobar Islands'
        else:
            temp_str
        temp_str = state_list[i].replace(" ", "+")
        line = str(get_lat_long(temp_str, state_list[i]))
        time.sleep(15)
        jk_line1 = line.replace("[", "")
        jk_line2 = jk_line1.replace("]", "")
        jk_line2 = jk_line2.replace("'", "")
        jk_line2 = jk_line2.replace('"', "")
        f.write(jk_line2)
        f.write('\n')
```

In [490]: *# checking the shapes of data frames, before we plot.*

```
#plot_crime_data.shape
t_final_frame = pd.read_csv('./crime-in-india/external/state_ut_lat_lo
n.csv')
t_final_frame.shape
```

Out[490]: (34, 5)

In [950]: *# Concatenating the data frames before plotting.*

```
plot_crime_data.reset_index(drop=True, inplace=True)
t_final_frame.reset_index(drop=True, inplace=True)

dp = pd.concat( [plot_crime_data, t_final_frame], axis=1)
```

In [951]: dp.head()

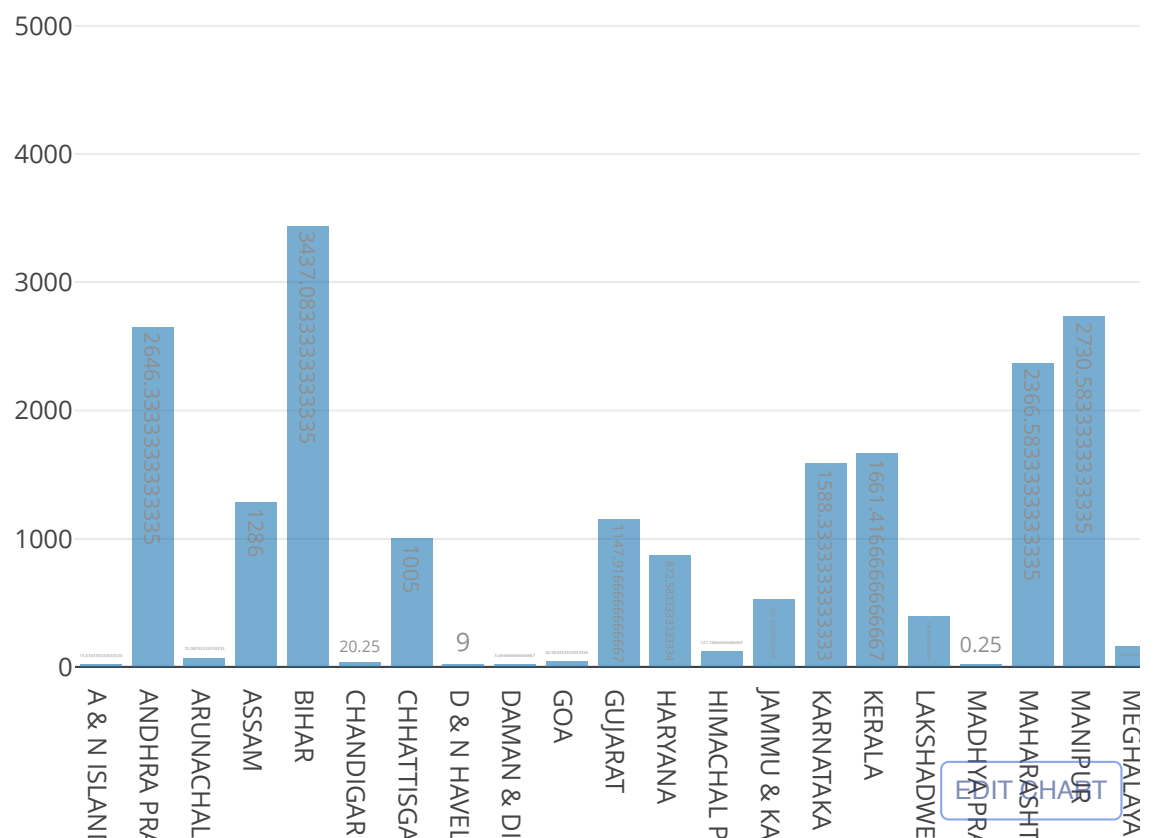
Out[951]:

	MURDER	Murder_Marker_Value	RAPE	Rape_Marker_Value	STATE/UT
0	13.333333	0.133333	9.083333	0.090833	A & N ISLANDS
1	2646.333333	26.463333	1123.250000	11.232500	ANDHRA PRADESH
2	70.083333	0.700833	41.666667	0.416667	ARUNACHAL PRADESH
3	1286.000000	12.860000	1348.166667	13.481667	ASSAM
4	3437.083333	34.370833	1093.666667	10.936667	BIHAR

In [964]: *# Plotting the indicators in bar to understand how the mean values are distributed*

```
data = [Bar(x = dp.place, y = dp.MURDER, text=dp.MURDER, textposition = 'auto',opacity=0.6, )]
py.iplot(data, title='basic bar chart')
```

Out[964]:



In [953]: *# Plotting the prominent spots along with Indicators*

```
mapbox_access_token = ' '
```

```
df = pd.read_csv('./crime-in-india/external/np_lat_long.csv')
site_lat = df.lat
site_lon = df.lon
locations_name = df.place
```

```
#Indicator 1 - Murder
```

```
ind1_lat = dp.lat
ind1_lon = dp.lon
ind1_loc_name = "Indicator - MURDER"
ind1_sz = dp.Murder_Marker_Value

#Indicator 2 - Rape
ind2_lat = dp.lat
ind2_lon = dp.lon
ind2_loc_name = "Indicator - RAPE"
ind2_sz = dp.Rape_Marker_Value

data = Data([
    Scattermapbox(
        lat=site_lat,
        lon=site_lon,
        mode='markers',
        name='Prominent Tourist Spots',
        marker=Marker(
            size=12,
            color='rgb(255, 0, 0)',
            opacity=0.8
        ),
        text=locations_name,
        hoverinfo='text'
    ),
    Scattermapbox(
        lat=ind1_lat,
        lon=ind1_lon,
        mode='markers',
        name='Indicator - Murder',
        marker=Marker(
            size=ind1_sz+50,
            color='rgb(0, 0, 255)',
            opacity=0.4
        ),
        text=ind1_loc_name,
        hoverinfo='text'
    ),
    Scattermapbox(
        lat=ind2_lat,
        lon=ind2_lon,
        mode='markers',
        name='Indicator - Rape',
        marker=Marker(
            size=ind2_sz+50,
            color='rgb(0, 102, 0)',
            opacity=0.5
        ),
        text=ind2_loc_name,
        hoverinfo='text'
    )
])
```

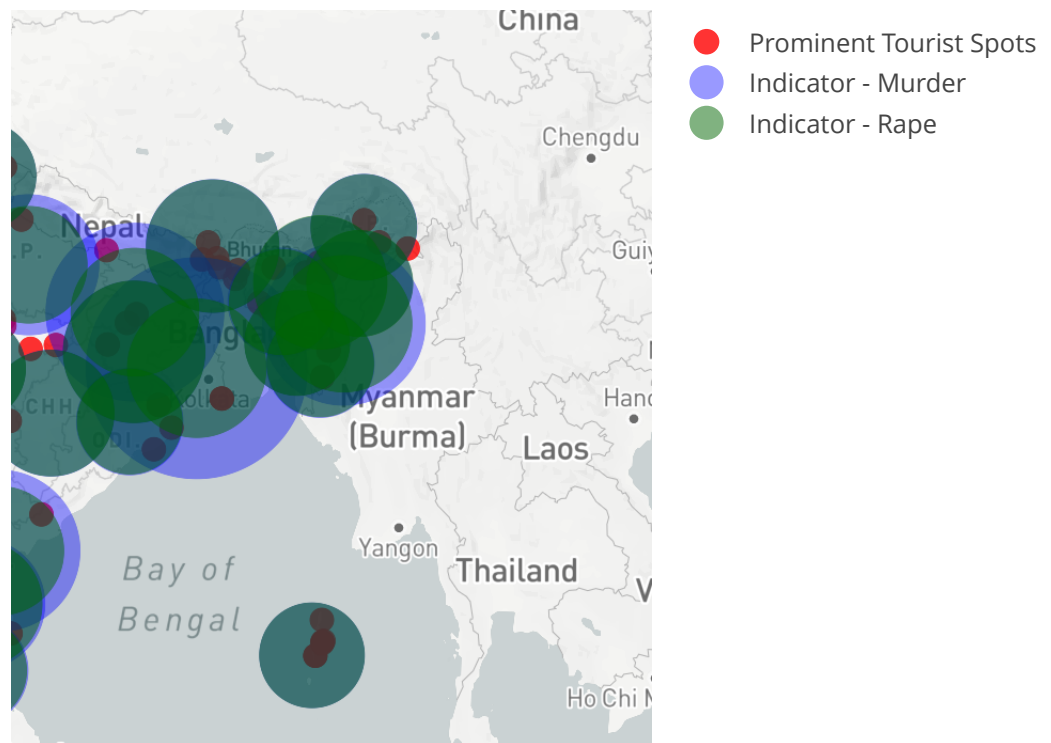
```
    )]
)

layout = Layout(
    title='Prominent Tourist Spots in India - Along with Indicators',
    autosize=True,
    hovermode='closest',
    showlegend=True,
    mapbox=dict(
        accesstoken=mapbox_access_token,
        bearing=0,
        center=dict(
            lat=22.6,
            lon=93.4
        ),
        pitch=0,
        zoom=3,
        style='light'
    ),
)

fig = dict(data=data, layout=layout)
py.iplot(fig, filename='legend-names')
```

Out[953]:

Prominent Tourist Spots in India - Along with Indicators

[EDIT CHART](#)

This is an interactive map. More importantly, comparing this with the distribution (bar chart) gives more insight to understand how crimes are distributed across locations

Understanding Police Force Available

Initial data exploration


```
In [684]: pl_data = pd.read_csv('./crime-in-india/Crime_data/12_Police_strength_
actual_and_sanctioned.csv')
pl_data.head()
```

Out[684]:

	Area_Name	Year	Group_Name	Sub_Group_Name	Rank_All_Ranks_Total	Rank_A5
0	Andaman & Nicobar Islands	2001	Actual Police Strength - Armed Police	A2. Actual Armed Police (Incl. Women Police)	766	7
1	Andhra Pradesh	2001	Actual Police Strength - Armed Police	A2. Actual Armed Police (Incl. Women Police)	12510	433
2	Arunachal Pradesh	2001	Actual Police Strength - Armed Police	A2. Actual Armed Police (Incl. Women Police)	2232	14
3	Assam	2001	Actual Police Strength - Armed Police	A2. Actual Armed Police (Incl. Women Police)	23963	36
4	Bihar	2001	Actual Police Strength - Armed Police	A2. Actual Armed Police (Incl. Women Police)	373	0

```
In [685]: # Let's understand the groups present.
# For this project, we will select Actual Vs Sanctioned Police Strength
h

pl_data['Group_Name'].unique().tolist()
```

```
Out[685]: ['Actual Police Strength - Armed Police',
'Actual Police Strength - Civil Police',
'Actual Police Strength - Total',
'Actual Women Police Strength - Armed Police',
'Actual Women Police Strength - Civil Police',
'Actual Women Police Strength - Total',
'Sanctioned Police Strength - Armed Police',
'Sanctioned Police Strength - Civil Police',
'Sanctioned Police Strength - Total',
'Sanctioned Women Police Strength - Armed Police',
'Sanctioned Women Police Strength - Civil Police',
'Sanctioned Women Police Strength - Total']
```

```
In [687]: list(pl_data)
```

```
Out[687]: ['Area_Name',
           'Year',
           'Group_Name',
           'Sub_Group_Name',
           'Rank_All_Ranks_Total',
           'Rank_ASI_Equivalent',
           'Rank_ASPDySPAssttCommandant',
           'Rank_Below_HC_and_Above_Constables',
           'Rank_Constables',
           'Rank_DGAddl_DG',
           'Rank_DIG',
           'Rank_Head_Constables',
           'Rank_IGSplIG',
           'Rank_Inspectors_Equivalent',
           'Rank_SI_Equivalent',
           'Rank_SSPSPAddlSPCommandant']
```

```
In [704]: # Removing the columns we don't need. We are keeping the total columns
           along with indicators and removing others.
```

```
tlist = [<Columns in List>]

for i in range(len(tlist)):
    del pl_data[tlist[i]]
```

```
In [802]: pl_data.head()
           pl_filter = pl_data['Group_Name'].str.contains('Total')
           pl_filter.head()
           pl_data.shape
```

```
Out[802]: (4188, 5)
```

```
In [803]: n_frame = pl_data[pl_filter]
n_frame.head()
#n_frame.shape
```

Out[803]:

	Area_Name	Year	Group_Name	Rank_All_Ranks_Total	place
698	Andaman & Nicobar Islands	2001	Actual Police Strength - Total	2792	Andaman & Nicobar Islands
699	Andhra Pradesh	2001	Actual Police Strength - Total	77846	Andhra Pradesh
700	Arunachal Pradesh	2001	Actual Police Strength - Total	5398	Arunachal Pradesh
701	Assam	2001	Actual Police Strength - Total	53345	Assam
702	Bihar	2001	Actual Police Strength - Total	45404	Bihar

```
In [804]: #n_frame['Rank_All_Ranks_Total'].unique().tolist()
n_frame.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1396 entries, 698 to 4187
Data columns (total 5 columns):
Area_Name      1396 non-null object
Year           1396 non-null int64
Group_Name     1396 non-null object
Rank_All_Ranks_Total  1396 non-null int64
place          1396 non-null object
dtypes: int64(2), object(3)
memory usage: 65.4+ KB
```

```
In [825]: # Creating actual data frames

act_filter = pl_data['Group_Name'].str.contains('Actual Police Strengt
h')
sanct_filter = pl_data['Group_Name'].str.contains('Sanctioned Police S
trength')

#aframe = pd.DataFrame(n_frame[act_filter])
#aframe.head()
aframe.info()
#del aframe['Year']

<class 'pandas.core.frame.DataFrame'>
Int64Index: 349 entries, 698 to 1046
Data columns (total 3 columns):
Area_Name                349 non-null object
Rank_All_Ranks_Total     349 non-null int64
place                    349 non-null object
dtypes: int64(1), object(2)
memory usage: 10.9+ KB
```

```
In [872]: # Creating a frame with mean values for the indicator - "Actual Police
Strength - Total"

a_frame = aframe.groupby('place').mean()
a_frame['place']=aframe['place'].unique().tolist()
a_frame.head()
a_frame['Actual_Mean'] = a_frame['Rank_All_Ranks_Total']
a_frame.head()
```

Out[872]:

	Rank_All_Ranks_Total	place	Sanctioned_Mean
place			
Andaman & Nicobar Islands	2971.4	Andaman & Nicobar Islands	2971.4
Andhra Pradesh	84085.6	Andhra Pradesh	84085.6
Arunachal Pradesh	6146.8	Arunachal Pradesh	6146.8
Assam	52443.1	Assam	52443.1
Bihar	54630.9	Bihar	54630.9

In [868]: *# Creating Sanctioned data frames*

```
sanct_filter = pl_data['Group_Name'].str.contains('Sanctioned Police S  
trength')  
sframe = n_frame[sanct_filter]  
sframe.shape
```

Out[868]: (349, 3)

In [871]: *# Creating a frame with mean values for the indicator - "Sanctioned Po
lice Strength - Total"*

```
s_frame = sframe.groupby('place').mean()  
s_frame['place']=sframe['place'].unique().tolist()  
s_frame.head()  
s_frame['Sanctioned_Mean'] = s_frame['Rank_All_Ranks_Total']  
s_frame.head()
```

Out[871]:

	Rank_All_Ranks_Total	place	Sanctioned_Mean
place			
Andaman & Nicobar Islands	3219.1	Andaman & Nicobar Islands	3219.1
Andhra Pradesh	93663.6	Andhra Pradesh	93663.6
Arunachal Pradesh	6610.0	Arunachal Pradesh	6610.0
Assam	55950.4	Assam	55950.4
Bihar	75673.3	Bihar	75673.3

```
In [913]: #dx = pd.concat( [a_frame, s_frame], axis=1)
dx.head()
```

Out[913]:

	place	Actual_Mean	place	Sanctioned_Mean
place				
Andaman & Nicobar Islands	Andaman & Nicobar Islands	2971.4	Andaman & Nicobar Islands	3219.1
Andhra Pradesh	Andhra Pradesh	84085.6	Andhra Pradesh	93663.6
Arunachal Pradesh	Arunachal Pradesh	6146.8	Arunachal Pradesh	6610.0
Assam	Assam	52443.1	Assam	55950.4
Bihar	Bihar	54630.9	Bihar	75673.3

```
In [937]: dx['Occupancy(%)'] = (dx['Actual_Mean']/dx['Sanctioned_Mean'])*100
dx['Vacancy(%)'] = 100 - dx['Occupancy(%)']
dx.head()
```

Out[937]:

	place	Actual_Mean	Sanctioned_Mean	Occupancy(%)	Vacancy(%)
place					
Andaman & Nicobar Islands	Andaman & Nicobar Islands	2971.4	3219.1	92.3053	7.6947
Andhra Pradesh	Andhra Pradesh	84085.6	93663.6	89.774	10.226
Arunachal Pradesh	Arunachal Pradesh	6146.8	6610	92.9924	7.00756
Assam	Assam	52443.1	55950.4	93.7314	6.26859
Bihar	Bihar	54630.9	75673.3	72.1931	27.8069

```
In [942]: dx = dx.sort_values('place')
dx
```

Out[942]:

	place	Actual_Mean	Sanctioned_Mean	Occupancy(%)	Vacancy(%)

place					
Andaman & Nicobar Islands	Andaman & Nicobar Islands	2971.4	3219.1	92.3053	7.6947
Andhra Pradesh	Andhra Pradesh	84085.6	93663.6	89.774	10.226
Arunachal Pradesh	Arunachal Pradesh	6146.8	6610	92.9924	7.00756
Assam	Assam	52443.1	55950.4	93.7314	6.26859
Bihar	Bihar	54630.9	75673.3	72.1931	27.8069
Chandigarh	Chandigarh	4181.2	4710.6	88.7615	11.2385
Chhattisgarh	Chhattisgarh	26455.4	31085.4	85.1055	14.8945
Dadra & Nagar Haveli	Dadra & Nagar Haveli	219.9	244.2	90.0491	9.95086
Daman & Diu	Daman & Diu	231.5	242.8	95.346	4.65404
Delhi	Delhi	58933.5	65622.8	89.8064	10.1936
Goa	Goa	4049.5	4613.6	87.7731	12.2269
Gujarat	Gujarat	63171.5	74014	85.3507	14.6493
Haryana	Haryana	37760.1	47323.9	79.7908	20.2092
Himachal Pradesh	Himachal Pradesh	12343.5	14953.5	82.5459	17.4541
Jammu & Kashmir	Jammu & Kashmir	65096.6	69564.5	93.5773	6.42267
West Bengal	Jharkhand	81922.3	96158	85.1955	14.8045
Jharkhand	Karnataka	31900.6	36219.6	88.0755	11.9245
Karnataka	Kerala	58996.5	69342.2	85.0802	14.9198
Kerala	Lakshadweep	42604.9	46542.2	91.5404	8.45963
Lakshadweep	Madhya Pradesh	305.7	410.7	74.4339	25.5661
Madhya Pradesh	Maharashtra	75006.9	77244.9	97.1027	2.89728
Maharashtra	Manipur	151992	167715	90.6251	9.3749

Manipur	Meghalaya	15876.1	19895.7	79.7966	20.2034
Meghalaya	Mizoram	9319.2	11498.3	81.0485	18.9515
Mizoram	Nagaland	7923.7	8876.8	89.263	10.737
Nagaland	Odisha	11168.8	11452	97.5271	2.47293
Odisha	Puducherry	38061.3	44419.3	85.6864	14.3136
Puducherry	Punjab	2146.5	2513.4	85.4022	14.5978
Punjab	Rajasthan	69150.9	72269.7	95.6845	4.3155
Rajasthan	Sikkim	65907.7	73439.4	89.7443	10.2557
Sikkim	Tamil Nadu	3482.7	4010.2	86.846	13.154
Tamil Nadu	Tripura	85156.5	99220.1	85.8259	14.1741
Tripura	Uttar Pradesh	19705.3	24295.7	81.1061	18.8939
Uttar Pradesh	Uttarakhand	151254	205997	73.4252	26.5748
Uttarakhand	West Bengal	13702.7	15671.5	87.4371	12.5629

In [945]: *# Plotting the trend using plotly*

```

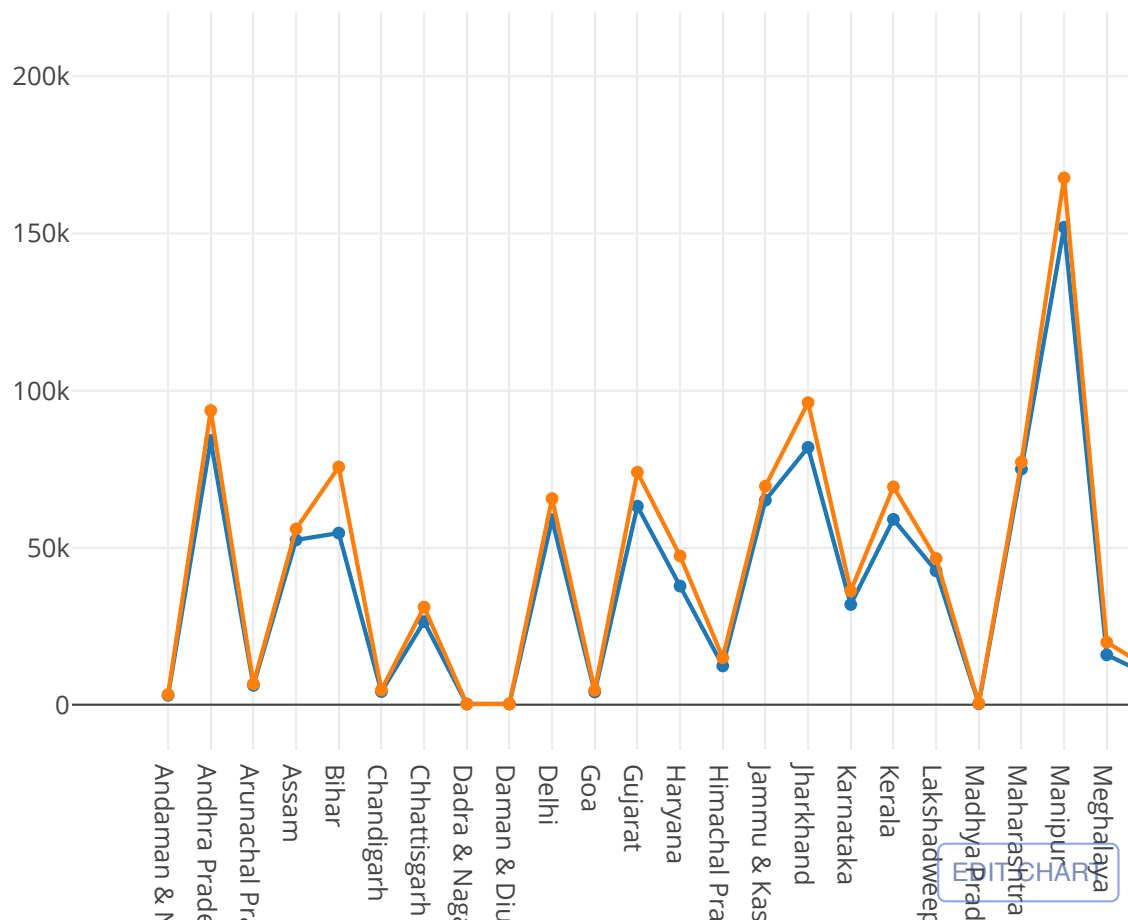
val_x = dx.place
act_y = dx.Actual_Mean
sct_y = dx.Sanctioned_Mean
trace = Scatter(
    x = val_x,
    y = act_y,
    mode = 'lines+markers',
    name = 'Actual')
tracel = Scatter(
    x = val_x,
    y = sct_y,
    mode = 'lines+markers',
    name = 'Sanctioned')

data = [trace, tracel]
layout = dict(title='Actual vs Sanctioned Police Force (mean value)')
fig = dict(data=data, layout=layout)
#trace = Scatter(x,y)
py.iplot(fig,file_name='line')

```


Out[945]:

Actual vs Sanctioned Police Force (



```
In [959]: pp = pd.read_csv('./crime-in-india/external/tabula-police_person_table.csv', encoding='Latin-1')
pp.head()
```

Out[959]:

	States / UTs.	Sanctioned	Actual
0	Andhra Pradesh	875	1,044
1	Arunachal Pradesh	102	120
2	Assam	605	710
3	Bihar	839	1,103
4	Chhattisgraph	392	465

```
In [960]: val_x = pp['States / UTs.']
act_y = pp.Actual
sct_y = pp.Sanctioned
trace = Scatter(
    x = val_x,
    y = act_y,
    mode = 'lines+markers',
    name = 'Actual')
tracel = Scatter(
    x = val_x,
    y = sct_y,
    mode = 'lines+markers',
    name = 'Sanctioned')

data = [trace, tracel]
layout = dict(title='Population per Policeman')
fig = dict(data=data, layout=layout)
#trace = Scatter(x,y)
py.iplot(fig,file_name='line')
```

Out[960]:

Population per Policema

